How to / Maniaplanet / Manialink / Build a Manialink

Author : Marcel @ Funtrackers

3. Manialink Elements

This chapter now will give you a complete reference about all elements of the ManiaPlanet ManiaLinks, including a detailled list of the supported attributes. If you are new to the ManiaLinks at all, you may want to read once through this part of the tutorial, to get a little overview of which elements exist and for what each of them is used for.
The intended purpose, nevertheless, is a reference: You do not have to remember all details, there is no test in the end to prove your skills.

<manialink>

The <manialink> tag is the godfather of all tags, it surrounds as root node always all other elements of the ManiaLink. If a ManiaLink does not start with the <manialink> tag, it simply is no ManiaLink.

Attributes:

- **version="1"** – The version of the ManiaLink
  With version you are able to specify the version of the ManiaLink. Depending on this version, the ManiaLink may be displayed slightly different and other features may be available for it. ManiaLinks from TrackMania Forever uses the version 0, which is at the same time the default value of this attribute, whereas ManiaLinks from ManiaPlanet uses the version 1. It is recommended to use version 1 on all new developped ManiaLinks, which should not be compatible to TrackMania Forever, to enable all features of the ManiaPlanet ManiaLinks. See ManiaLink Version 1 for more information.
- **background="0"** – The visibility of the background
  If background is set to 0, the ManiaLink won't have any background itself, but will use whatever the game shows up behind the ManiaLink. So this feature can be seen as a kind of "transparent background".

Example

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<manialink version="1">
<!-- Content of the ManiaLink -->
</manialink>
```

<timeout>

To save some traffic, all requested ManiaLinks get locally cached by ManiaPlanet, until the game is quit. If now one of these cached ManiaLinks is called another time, ManiaPlanet will used the local copy of the file instead of re-requesting it again from the server. Which may be a nice features for static ManiaLinks is a huge problems for dynamic ones: Last changes will never display, as ManiaPlanet does not reload the file from the server.

With the <timeout> tag you can disable these caching of the ManiaLink files: The value specified between the tags says the number of seconds, after which the game should re-request the file from the server. Typically this value is set to 0 to force ManiaPlanet to always reload the file from the server.

Example:
Skeleton of a ManiaLink including the <timeout> tag:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<manialink version="1">
<timeout>0</timeout>
<!-- Content of the ManiaLink -->
</manialink>
```

<frame>

A Frame is for grouping several elements together, which got written between the <frame> tags. The Frame itself is not visible, but with its position and scaling factor it influences all containing elements in their display. This means: If the position of the Frame is changed, all elements in this Frame will move along to this change.

Frames should always be used when several elements of the ManiaLink belongs logically together. For example: If you group all elements of the menu together in a Frame, moving the whole menu a bit down means only to change the position in the Frame; the actual Quads and Labels have not to be changed at all. In addition, you may interlace a Frame into another one. This gives you the possibility to build a kind of hirarchical structure within your ManiaLink.

Attributes

- pos(n)="X Y Z" – The position of the Frame
- scale="factor" – The scaling factor of the Frame
- id="identifier" – The ID of the Frame for the ManiaScript – (Class CGameManialinkFrame)

Example
A Frame to logically combine several Quads:

```
<frame posn="50 50 0">
<quad sizen="20 20" bgcolor="F00A" />
<quad posn="-20 0 0" sizen="20 20" bgcolor="00FA" />
</frame>
```

This example shows a Frame containing two Quads. These Quads have been positioned directly next to each other, and aligns with the Frame: If we move the Frame, the Quads will do so, too, but they will never get disconnected in this way.

<format>

With the <format> tag you are able to define some attributes to be used on other elements as a default format. These predefined formats are applied to all elements of the current Frame in which the <format> is placed and all its subframes. This means, if you place the <format> tag as a direct child into the <manialink> tag, these formats are applied to all elements of the ManiaLink.

The <format> definitions affects the elements <quad>, <label>, <entry> and <fileentry>, whereas the predefined format always can be overwritten by specifying the same attribute directly in the element's tag.

Attributes

- bgcolor="RGBA" – The background color of the Element – Affects: Quad
  The colors are specified as a four digit hexadecimal number, having the values 0 to 9 and A to F for each digit. The first three digits represent the actual color in their red, green and blue components, and the last digit the transparency of the color, ranging from 0 for transparent to F for opaque.
- textsize="Number" – The text size of the Element – Affects: Label, Entry, FileEntry
- textcolor="RGBA" – The text color of the Element – Affects: Label, Entry, FileEntry
- focusareacolor1="RGBA" – The background color of the Element (without MouseOver) – Affects: Label, Entry, FileEntry
- focusareacolor2="RGBA" – The background color of the Element (with MouseOver) – Affects: Label, Entry, FileEntry
  Both attributes focusareacolor1 and focusareacolor2 define the background color of the element. Thereby focusareacolor1 defines the color to be used when the mouse is not over the element, and focusareacolor2 defines the color when the mouse is currently over it.
- style="StyleName" – The predefined style of the Element – Affects: Label, Entry, FileEntry
- With the style attribute you are able to use the same styles as ManiaPlanet uses in its own menus. For example, with style="CardButtonSmall" you can create the black button which gets white when moving the mosue over it. A list of available styles can be viewed on the ManiaLink Exemple.

Example
Example of using a <format> tag:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<manialink version="1.0">
<frame>
<format bgcolor="F00A" />
<quad sizen="10 10" />
<quad posn="10 0" sizen="10 10" bgcolor="0F0A" />
</frame>
<quad posn="20 0" sizen="10 10" />
</manialink>
```

The <format> sets the default background color to red, and this color gets applied to the first Quad. The second Quad defines a background color for its own, and therefore overwrites the predefined value of the <format>. The third and last Quad is not visible at all, because it is outside the <frame> and so the predefined value of the <format>is not applied to it.

<quad>

One of the most important elements of the ManiaLinks is the Quad for displaying images and shapes. It is possible to use the designs of ManiaPlanet itself, so called styles, to better integrate your ManiaLink into the game. Furthermore it is possible to link a <quad> to another ManiaLink or to an external website, whereas this linking can be highlighted when the mouse is moved over the Quad to indicate, that it is linked somewhere. As images a <quad> accepts the formats .jpg, .png, .tga and .dds.

Attributes

- pos(n)="X Y Z" – The Position of the Quad
- size(n)="width height" – The size of the Quad
- scale="factor" – The scaling factor of the Quad
- halign="left|center|right" – The horizontal alignment of the Quad
- valign="top|center|bottom" – The vertical alignment of the Quad
- style="StyleName" – The category of the predefined style of the Quad
- substyle="SubstyleName" – The concrete predefined style of the Quad
  To use a predefined style from the game, both attributes style and substyle have to be specified, where style can be seen as the category of the concrete style to be used. A list of all available styles can be viewed on the ManiaLink Exemple.
- image="image.png" – The image of the Quad
- imagefocus="image.png" – The image of the MouseOver effect of the Quad
  You may specify the images to be used for image and imagefocus either with their full URL like http://localhost/image.jpg or as URL relative to the XML file like ./image.jpg. If the mouse is moved into the Quad, the image specified with image is replaced with that from imagefocus if the latter has been specified and the Quad has been linked.
- bgcolor="RGBA" – The background color of the Quad
  As style/substyle, image/imagefocus and bgcolor all affect the Quad in how it is displayed, you cannot use all these attributes at the same time. In detail the style overwrites the images and both overwrites the background color.
- manialink="ManiaLink" – Link to another ManiaLink via its registered code or URL
- url="website.html" – Link to an external website
  To display an external website, ManiaPlanet minimizes itself and opens the default browser of the system.
- id="identifier" – The ID of the Quad for the ManiaScript – Class CGameManialinkQuad
- ScriptEvents="1" – Enables mouse events to be triggered for the Quad

Example:
Example of Quad definitions:

```
<quad sizen="50 10" style="Bgs1InRace" substyle="BgWindow1" />
<quad posn="0 -60" sizen="50 10"
image="http://localhost/menu.png"
imagefocus="http://localhost/menuHover.png"
manialink="FunTrackers" />
```

This example shows two Quads, where the first one uses a predefined style of ManiaPlanet and the second one custom images.

<label>

Next to the Quads, Labels are the most important element of ManiaLinks to display any kind of texts. This is not limited to a single line of text what the name "label" may imply, but also for complex texts with line breaks. The text to be displayed may either be specified as attribute text or to be written between the <label> tags.

A special feature of the label (or a curse, whether you want to have it or not) is the automatic translation of certain words and word groups, which exists in the language files of ManiaPlanet. For example, if you write statistics into a label (as the only word), this will be translated to Statistik if a German user views the ManiaLink. If you don't want to have these words translated, simply add another space at the end of the text.

Attributes

- pos(n)="X Y Z" – The Position of the Label
- size(n)="width height" – The size of the Label
- scale="factor" – The scaling factor of the Label
- halign="left|center|right" – The horizontal alignment of the Label
- valign="top|center|center2|bottom" – The vertical alignment of the Label
- text="LabelText" – The Text to be displayed in the Label
  You may use any $-formats of ManiaPlanet in the text. As already mentioned, you may also specify the text between the <label> tags instead of using the text attribute.
- style="StyleName" – The predefined style of the Label
  In Labels specifying the attribute style already enables the use of a predefined style of the game. A list of available styles can be viewed on the ManiaLink Exemple.
- textsize="Number" – The text size of the Label
- textcolor="RGBA" – The text color of the Label
  As style and textsize/textcolor both affect the Label, exactly the text of it, in how it is displayed, you cannot use both of them at the same time. In detail the style will overwrite the text specific attributes.
- focusareacolor1="RGBA" – The background color of the Label (without MouseOver)
- focusareacolor2="RGBA" – The background color of the Label (with MouseOver)
  Both attributes focusareacolor1 and focusareacolor2 define the background color of the Label. Thereby focusareacolor1 defines the color to be used when the mouse is not over the Label, and focusareacolor2 defines the color when the mouse is currently over it. As style and focusareacolor1/focusareacolor2 both affect the Label, exactly the background of it, in how it is displayed, you cannot use both of them at the same time. In detail the style will overwrite the background specific attributes.
- autonewline="1" – Enables the automatic line breaking of the Label
  If autonewline is set to 1 long lines will be automatically breaked to fit the width of the label. Otherwise only line breaks entered into the XML file itself will be used as line break in the ManiaLink, having maybe lines to long for the Label.
- maxline="Number" – Limits the maximum number of lines displayed in the Label
  With maxline you may set a limit of lines to be displayed in the Label. All lines exceeding this limit will bi ignored and not displayed in the Label at all.
- manialink="ManiaLink" – Link to another ManiaLink via its registered code or URL
- url="website.html" – Link to an external website
  To display an external website, ManiaPlanet minimizes itself and opens the default browser of the system.
- id="identifier" – The ID of the Label for the ManiaScript – Class CGameManialinkLabel
- ScriptEvents="1" – Enables mouse events to be triggered for the Label

Example
Example Label using a predefined style and $-formats:

```
<label style="CardButtonMediumWide" text="A button with a $F00red$g word" />
```

<audio>
As the name already says, you are able to add an Audio file to be played on your ManiaLink using the <audio> tag. This file is displayed as simple start and stop button, with which the user may stop or start the play of the file. As file formats .ogg and .mux are accepted.

Attributes

- pos(n)="X Y Z" – The Position of the Audio
- size(n)="width height" – The size of the Audio
- scale="factor" – The scaling factor of the Audio
- halign="left|center|right" – The horizontal alignment of the Audio
- valign="top|center|bottom" – The vertical alignment of the Audio
- data="audio.ogg" – The audio file to be played
- play="1" – Enables the automatic start of the Audio
  If play is set to 1 the audio file will start to play when the ManiaLink has finished loading.
- looping="0" – Disables the looping of the Audio
  If looping is set to 0 ManiaPlanet will stop playing the audio file after it finished once. Otherwise, ManiaPlanet will repeat it in an endless loop.
- id="identifier" – The ID of the Audio for the ManiaScript – Class CGameManialinkControl
- ScriptEvents="1" – Enables mouse events to be triggered for the Audio

Example
Example use of an <audio> tag:

```
<audio data="http://localhost/audio.ogg" play="1" />
```

<music>
The <music> tag is, too, used to play audio files. Instead of having the user control the play of the file as with the <audio>, <music> plays the file in the background without any possibility for the user to stop it in an endless loop. Again only .ogg and .mux files are accepted in this tag. As an additional feature the music does not stop playing if you change the ManiaLink, if the new one uses the very same audio file in its <music> tag.

Hint: The <music> tag only works if it is placed as a direct child of the <manialink> tag, i.e. outside of all Frames.

Attributes

- data="music.ogg" – The music file to be played

Example
Example of a <music> tag with its only attribute:

```
<music data="http://localhost/music.ogg" />
```

<include>
The name of the tag <include> already says, that it will include another XML file into the current ManiaLink. This enables the possibility to extract often used elements like the menu into a separate file, and to include it in all actual ManiaLinks: If you want to change the menu, you only have to change the separate file rather than change all files which uses this menu.

The inclusion of other files is limited to one level. Thus, you can only include a file from within the "main" file, but you are not able to include another file in the included file. The reason is to avoid a cycle of inclusions: If file A includes file B, and file B now is able to include file A, you get an endles recursion of included files.

Another point is, that the included file has to be a complete and valid XML file, and not a complete ManiaLink. This means, the included file must have an XML header and a single root node (mostly a <frame> which contains the actual elements to be included), and must not have a <manialink> tag itself. You can imagine the including process like cutting the main ManiaLink into two pieces, and pasting in the file to be included. Having an additional <manialink> at this point would make no sense at all.

Hint: Please keep in mind, that including a file using an <include> tag increases the time until the ManiaLink has been completed loading, as the included file has to be requested separatly from the server, so do not overuse this tag. If you have a dynamic (i.e. PHP based) ManiaLink, you should always consider to prefer the include or require commands of PHP instead.

Attributes

- url="include.xml" – The URL of the file to be included

Example

Main ManiaLink file:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<manialink version="1.0">
<quad sizen="10 10" bgcolor="F00A" />
<include url="file://Media/Manialinks/include.xml" />
</manialink>
```

File include.xml to be included:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<frame>
<quad posn="10 0" sizen="10 10" bgcolor="0F0A" />
<quad posn="20 0" sizen="10 10" bgcolor="00FA" />
</frame>
```

The second code snippet shows the file to be included. As already mentioned, it does not have a <manialink> itself, but needs to have a <frame> surrounding the actual elements to be included, as a valid XML file only is allowed to have a single root node.

<entry>

Another possibility to interact with the user is to show an input field, called Entry in ManiaLinks, where he or she can input any value. Of course, an Entry only makes sense on dynamic ManiaLinks, where a script on the server (e.g. written in PHP) is able to handle the inputted value. The use of an <entry> is pretty simple: You specify an unique name, and wherever this name appears in a link of a Label or a Quad it is replaced with the current value of the Entry. You should remember, that an Entry always accepts string values, thus you have to check on server side, if for example you only want to have numbers as values.

Attributes

- pos(n)="X Y Z" – The Position of the Entry
- size(n)="width height" – The size of the Entry
- scale="factor" – The scaling factor of the Entry
- halign="left|center|right" – The horizontal alignment of the Entry
- valign="top|center|center2|bottom" – The vertical alignment of the Entry
- style="StyleName" – The predefined style of the Entry
- textsize="Number" – The text size of the Entry
- textcolor="RGBA" – The text color of the Entry
  As style and textsize/textcolor both affect the Entry, exactly the text of it, in how it is displayed, you cannot use both of them at the same time. In detail the style will overwrite the text specific attributes.
- focusareacolor1="RGBA" – The background color of the Entry (without MouseOver)
- focusareacolor2="RGBA" – The background color of the Entry (with MouseOver)
  Both attributes focusareacolor1 and focusareacolor2 define the background color of the Entry. Thereby focusareacolor1 defines the color to be used when the mouse is not over the Entry, and focusareacolor2 defines the color when the mouse is currently over it. As style and focusareacolor1/focusareacolor2 both affect the Entry, exactly the background of it, in how it is displayed, you cannot use both of them at the same time. In detail the style will overwrite the background specific attributes.
- name="EntryName" – The unique name of the Entry
  As already mentioned, the name of the Entry is very important: Wherever this name occurs in a link, it is replaced with the current value entered by the user.
- default="value" – The default value of the Entry
  When loading the ManiaLink, the Entry is initialized with the value specified with default. As long as the user does not enter any other value, this default value will be displayed in the Entry.
- autonewline="1" – Enables the automatic line breaking of the Entry
  If autonewline is set to 1, the entered value is automatically broken into a new line if it exceeds the set width of the Entry. This is helpful if you expect larger inputs from the user.
- id="identifier" – The ID of the Entry for the ManiaScript – Class CGameManialinkEntry
- ScriptEvents="1" – Enables mouse events to be triggered for the Entry

Example

Simple example of an Entry and a link with its value:

```
<entry sizen="50 5" style="TextValueSmall" name="inputValue" default="Hello World!" />
<label posn="0 -10" style="CardButtonMedium" text="Send Value" url="http://localhost/script.php?value=inputValue" />
```

This example shows an Entry with the name inputValue, and a Label which sends its value to a script: The inputValue of the link will be replaced with the current value of the Entry. Please refer to the Advanced Example of Entry and FileEntry for additional information.

<fileentry>

The <fileentry> works the same way as <entry> does, with the only difference that the user does not enter a value, but select a file to tbe uploaded to the server. Like the Entry, the FileEntry uses its unique name to Again, this tag only makes sense on dynamic ManiaLinks where a server script in the background can handle the uploaded file.

As soon as the user clicks into the field of FileEntry, a file selecting dialog will pop up, having the user to choose a file from his computer. This file then is transmitted using a POST request, see the Advanced Example of Entry and FileEntry for additional information.

Hint: Always check what file the user tries to upload, before finally saving it on the server. Never ever save any files without checking them, as this is a huge security risk!

Attributes

- pos(n)="X Y Z" – The Position of the FileEntry
- size(n)="width height" – The size of the FileEntry
- scale="factor" – The scaling factor of the FileEntry
- halign="left|center|right" – The horizontal alignment of the FileEntry
- valign="top|center|center2|bottom" – The vertical alignment of the FileEntry
- style="StyleName" – The predefined style of the FileEntry
- textsize="Number" – The text size of the FileEntry
- textcolor="RGBA" – The text color of the FileEntry
  As style and textsize/textcolor both affect the FileEntry, exactly the text of it, in how it is displayed, you cannot use both of them at the same time. In detail the style will overwrite the text specific attributes.
- focusareacolor1="RGBA" – The background color of the FileEntry (without MouseOver)
- focusareacolor2="RGBA" – The background color of the FileEntry (with MouseOver)
  Both attributes focusareacolor1 and focusareacolor2 define the background color of the FileEntry. Thereby focusareacolor1 defines the color to be used when the mouse is not over the FileEntry, and focusareacolor2 defines the color when the mouse is currently over it. As style and focusareacolor1/focusareacolor2 both affect the FileEntry, exactly the background of it, in how it is displayed, you cannot use both of them at the same time. In detail the style will overwrite the background specific attributes.
- name="FileEntryName" – The unique name of the FileEntry
  The name of a FileEntry has the same task as in an Entry: Wherever this name appears in a link, it is replaced with the name of the selected file of the FileEntry.
- folder="path" – The path of the base folder
  With the folder attribute you may help the user to find the wanted file to be uploaded to the ManiaLink, as you set the base directory the user is able to browse. This folder is always relative to the ManiaPlanet folder in the Documents directory, and the user is not allowed to browse any upper folders as the specified one. If for example you only want to have Maps uploaded to the ManiaLink, specify Maps as base folder, but keep in mind, that the user still is able to upload any file he or she wants: Double check on the server!
- default="value" – The default value of the FileEntry
  The value of the default attribute is displayed as long as the user did not choose a file from his computer.
- autonewline="1" – Enables the automatic line breaking of the FileEntry
  autonewline has the same effect as in an <entry>, but as filenames won't get very long in most cases, you won't need this attribute very often.
- id="identifier" – The ID of the FileEntry for the ManiaScript – Class CGameManialinkFileEntry
- ScriptEvents="1" – Enables mouse events to be triggered for the FileEntry

Example
Basic example of an FileEntry and how the file is send to a server script:

```
<fileentry sizen="50 5" style="TextValueSmall" name="inputFile" folder="Maps" default="Choose Map..." />
<label posn="0 -10" style="CardButtonMedium" text="Send File" url="POST(http://localhost/script.php?
file=inputFile,inputFile)" />
```

This example is pretty much the same as that one for <entry>, with the only different that we need to use the POST() method to link to another ManiaLink or website. Again, please refer to the Advanced Example of Entry and FileEntry to have a more detailed description for this.

<dico>

With the <dico> tag and some <language> tags, you are able to translate your ManiaLink into other languages, making it possible that an English speaking player sees an English version, whereas a German player has a German version. The actually used language is selected by the game: If the player's language (which he or she selected in the configuration of ManiaPlanet) is available, that one will be used, otherwise ManiaPlanet will try to use the English version. If both do not exist, nothing is displayed as translation, so ensure that at least the English language is complete.

Making a ManiaLink multilingual is very easy: Instead for example specifying the exact text in a <label>, you only use a ID for it and give for each language in which your ManiaLink should be available the translation. To illustrate it better, an example should be given:

Example:
A simple but multilingual ManiaLink:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<manialink version="1">
<timeout>0</timeout>
<!-- The ManiaLink elements itself -->
<label posn="-20 0" textid="example" />
<label posn="-20 -10" textid="thanks" />
<quad sizen="10 5" imageid="img" />
<-- The translations of the IDs -->
<dico>
<language id="en">
<example>Example</example>
<thanks>Thanks Nadeo for the new ManiaPlanet ManiaLinks!</thanks> <img>http://localhost/manialink/en.png</img>
</language>
<language id="de" >
<example>Beispiel</example>
<thanks>Danke Nadeo für die neuen ManiaPlanet ManiaLinks!</thanks> <img>http://localhost/manialink/de.png</img>
</language>
</dico>
</manialink>
```

So what we can see in the first part of the ManiaLink are the elements, which have been explained above. The only different is, that instead e.g. text the attribute textid is used. This tells ManiaPlanet, that it has to look up the specified ID in the <dico> tag and replace the ID with the actual translation. In fact, you simply have to add "id" to the name of the attribute, to enable the translation feature for it. The following attributes support to be multilingual:

List of supported attributes

- <quad> – image, imagefocus, url, manialink
- <label> – text, url, manialink
- <audio> – data
- <video> – data

The second part of the example shows the actual translation of the IDs, enclosed by the <dico> tag. For each language to be supported, add a <language> tag to the <dico>, holding as only attribute id the ID of the language, you want to translate. Which ID you have to use for which language shows the following list:

List of language codes

- cz – Czech

- da – Danish
- de – German
- en – English
- es – Spanish
- fr – French
- hu – Hungary
- it – Italian
- jp – Japanese
- kr – Korean
- nb – Norwegian
- nl – Dutch
- pl – Polish
- pt – Portuguese
- pt_BR – Brazilian Portugese
- ro – Romanian
- ru – Russian
- sk – Slovak
- tr – Turkish
- zh – Chinese

For each language, you now have to add all IDs to be translated as child tags to the <language>, as shown in the example. As the IDs to be translated become tag-names, it is best if you only choose IDs, which begin with a letter, and may be followed bay other letters, numbers or the underscore _. Try to avoid any special characters within these IDs, as they may invalidate the XML syntax, and thus making the whole ManiaLink file invalid. The translation itself, denoted between the ID-tags, may contain whatever character you want, but keep in mind that the characters "<", ">" and "&" have to be replaced by "<", ">" and "&" respectively, to keep the XML syntax valid. Also remember saving the XML file in UTF-8 as soon as you use special characters like German umlauts "Ä", "Ö", "Ü" etc.

<script>

The <script> tag is maybe the most simple tag, and at the same point the most powerful element of a ManiaLink: With this tag, you may include a ManiaScript into your ManiaLink. The use of it is very intuitive: The ManiaScript itself gets written between the <script> tags. But as some elements of the ManiaScript may invalidate the XML syntax, you have to put the ManiaScript additionally into an XML-comment.

It is possible, to use several <script> tags in your ManiaLink. These parts will get clued together on loading the ManiaLink, and executed as one big ManiaScript. So there is no different in using a ManiaScript in three parts, or to include it as one big part e.g. at the end of the ManiaLink.

For additional information, please look at the corresponding part of the Tutorial ManiaScript in ManiaLinks.

Example:
A simple example of a complete ManiaLink using a ManiaScript:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<manialink version="1">
<quad id="myQuad" posn="10 0" sizen="20 20" bgcolor="F00A" ScriptEvents="1" />
<script> <!--
while(True)
{
foreach(Event in PendingEvents)
{
if (Event.Type == CGameManialinkScriptEvent::Type::MouseClick)
{
Page.MainFrame.Controls["myQuad"].PosnX *= -1;
}
}
yield;
}
--></script>
</manialink>
```

At this point, I do not want to explain the ManiaScript in detail. Save this code as a new XML file, open it in the ManiaPlanet Browser, and click on the Quad.
This example should only demonstrate, how to integrate a ManiaScript into a ManiaLink: The ManiaScript itself has to be surrounded with an XML-comment, to keep the XML file itself valid.

This site is friendly to stationary (PC/MAC) and mobile (Android/IOS) browsers
Trackmania, Shootmania, Questmania games & media © Nadeo & UbiSoft used under fair use policy
8,373,036 page views since January 2008. All tutorials, services, coding © Mania Creative 2008-2013